

## **REMARKS**

Claims 1-32 are pending in the present application. By the current action, claims 5, 7-8, 13, 15-16, 21, 23-24, 26, 28, and 30 are amended. Reconsideration of the claims is respectfully requested.

### **I. Information Disclosure Statement**

It was noted in the outstanding office action that the submission of the Information Disclosure Statement did not contain a copy of two articles cited in the IDS. Copies of these articles are enclosed with the current response. It is noted that the article "*Graphics Interchange Format(sm), Version 89a*" is present on the Internet at a number of different sites. The enclosed copy was reformatted somewhat after downloading from the Internet, but was not otherwise changed. The website from which the article was taken is noted thereon. Additionally, the second article, from the IBM Technical Disclosure Bulletin, is the best copy that the undersigned agent was able to obtain.

### **II. Objection to Specification**

The abstract was objected to as not following proper guidelines. This objection is traversed. However, in response to the objection, the abstract has been re-worded and shortened.

Additionally, the Examiner cited several typographical errors in the specification. The Examiner is thanked for finding these errors, which have been corrected. No new matter has been added by any of the amendments to the specification.

### **III. Objection to Claims: Claims 5, 26, 28 and 30**

Claims 5, 26, 28 and 30 were each objected to; the reasons for the objections follow, as well as the response:

- Claims 5 and 26 were almost identically worded. In response to this objection, an additional feature has been added to claim 26, as is discussed in the arguments.
- Line 3 of claim 28 contained a typographical error. In response, the error has been corrected.

- The preamble of claim 30 was objected to as unclear, as the body of the claim claimed functionality that cannot be accomplished by the claimed data structure. In response to this rejection the Examiner's suggested correction was incorporated in the claim, with the minor change of reciting that, "the computer program product" comprises the details found in the body of the claim.

#### IV. **35 U.S.C. § 102, Anticipation: Claims 1-6, 9-14, 17-22, and 25-32**

Claims 1-6, 9-14, 17-22, and 25-32 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Cook *et al.*, Method and Apparatus for Creating Interactive Web Page Objects, U.S. Patent No. 6,178,432, January 23, 2001 (hereinafter "**Cook**"). This rejection is respectfully traversed.

The claims will be broken into several groups in order to discuss various differences in the recited features. It is generally noted that the cited figures and excerpts of **Cook** show that this patent can achieve an end result that shares some similarities with the end result achieved by the invention recited in the claims (i.e. both **Cook** and the present application show the display of a button being affected by a user activity). However, these results are achieved by different means, as will be discussed in greater detail below.

#### **Claims 1, 9, 17, 25, 27, and 30**

The rejection states:

Concerning claims 1, 9, 17, and 30, Cook describes a method, computer program product embodied in a machine-readable medium, and system comprising: a memory unit operable for associating graphical images displayed on a web page with client side events; and a processor coupled to said memory unit, wherein said processor, responsive to said computer program, comprises: circuitry operable for providing graphical image information including control information and rendering information, said control information for controlling a display of said rendering information, wherein said control information comprises an image identifier value; circuitry operable for receiving an event identifier value in response to a client initiated action; and displaying said rendering information in response to said event identifier value matching said image identifier value in said control information, wherein said rendering information represents at least one graphical image comprising differential information for generating a composite image. For example, Cook describes Figure 26 as a hardware setup used in one embodiment of his invention. He describes computer 250 as including processing unit 252 and storage, or memory, unit 256 column 6 lines 47-53. Stored within the memory unit is

module 208, which creates a webpage that is capable of associating displayed graphical images with client side events column 4 lines 44-66. The client side event has associated control information that controls what is rendered on the screen and where it is rendered column 10 lines 35-49. In this particular case the position of the mouse cursor during a client side event determines the control information sent to memory to decide which image or images are to be presented. The associated Java applet looks up all objects with the associated mouse position and determines what action, if any, to take on each object. Figures 1A and 1B offer a visual depiction of the method described. Figure 1A is the original presentation of a webpage, and Figure 1B is the composite image that results from a user clicking on various objects on the screen. All tasks are completed without refreshing the page column 4 lines 53-67 and column 5 lines 1-11,<sup>1</sup>

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994).

Claim 1, which exemplifies this group, recites:

1. (Original) A method comprising the steps of:
  - receiving graphical image information including control information and rendering information, said control information for controlling a display of said rendering information, wherein said control information comprises an image identifier value;
  - receiving an event identifier value in response to a client initiated action;
  - and
  - displaying said rendering information in response to said event identifier value matching said image identifier value in said control information, wherein said rendering information represents at least one image comprising differential image information for generating a composite image.

**Cook** does not anticipate the invention recited in exemplary claim 1 in that **Cook** does not show the step, “*wherein said rendering information represents at least one image comprising differential image information for generating a composite image*” (emphasis added). The rejection asserts that **Cook** shows this feature, stating:

Figures 1A and 1B offer a visual depiction of the method described. Figure 1A is the original presentation of a webpage, and Figure 1B is the composite image that results from a user clicking on various objects on the screen. All tasks are

---

<sup>1</sup> Office Action dated October 4, 2005, pages 4-5

completed without refreshing the page column 4 lines 53-67 and column 5 lines 1-11.<sup>2</sup>

Although the rejection thus asserts that the changes to the webpage in **Cook** are performed without refreshing the page, **Cook** itself does not support this assertion. Figure 4A of **Cook** is reproduced below.

The figure is described thus:

FIG. 4A is a flow diagram which illustrates a process which is executed by a Java applet downloaded by a user from Web server 210 in order to display and interact with an interactive Web page. The Java applet displays the objects according to the rules implicit in their defined structures and performs the actions specified in the behaviors on the targets upon the occurrence of the events specified in the behaviors. The actions implemented include primitive actions such as showing, hiding, or playing objects as well as actions which cause state changes in objects and notification actions which occur automatically as a result of the inclusion of an object within certain structures such as stacks or switches. Actions which cause state changes often trigger other actions which are referred to as "triggered actions." Triggered actions are defined by behaviors so that they occur in response to state changes or other events which are caused indirectly by behaviors of objects instead of being caused directly by actions of the user.

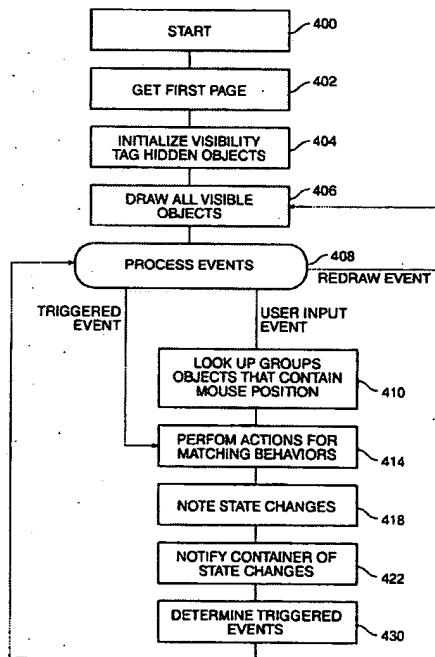


FIG. 4A

The process starts at step 400 and in step 402, the applet gets the first "page". The "page" is not a standard Web page but is instead a set of objects and/or structures which include behaviors. These objects and structures are not grouped together in order to be displayed simultaneously on a single web page, but are instead grouped together to be displayed interactively on an interactive "page" in response to input events caused by the user. In step 404, the applet initializes the visibility of each object. Objects which are to be hidden are tagged as hidden. For example, every object in each stack except for the first object is tagged as hidden. All visible objects are drawn in step 406. Visible objects are said to be "drawn" when they are bit mapped onto the display. In other embodiments, displays which are not bit mapped displays are used. In such embodiments, visible objects are drawn when they are presented on the display. In general, objects are described as being either "drawn" or

<sup>2</sup> Office action of 10/04/2005, page 5, lines 12-16, emphasis added

"displayed" when they are made visible (i.e. not tagged as being hidden). Objects which are drawn are also referred to as being "played." Being "played" is a more general term which refers to shape objects and other multimedia objects such as sound objects which become active. Next, the applet enters a state 408 in which it waits for events and processes those events as they occur.

A redraw event is a special event which is generated by the system. The screen bit map is regenerated from the objects which are visible. Upon the occurrence of a redraw event, control is transferred to step 406 and all visible objects, that is all objects which are not tagged as hidden, are redrawn. Control is then transferred back to state 408 and the system waits for other events.

If the event is a user input event, then the applet must determine the objects which contain behaviors that will be triggered by the event. In one embodiment, the user input device is a mouse and so the applet will look up all objects that contain the mouse position which is associated with the user input event in step 410. Events involving the mouse may include any movement of the mouse, the pressing of the mouse button ("mouse down"), the releasing of the mouse button ("mouse up"), a combination of a mouse down and a mouse up ("mouse click"), or any combination or sequence of mouse clicks. Special drag and drop events which also involve the mouse include the beginning of a drag ("drag start"), the ending of a drag ("drag end") and a drop. In some embodiments, the mouse has several buttons and an additional set of mouse events is defined for each button.

Once the applet has determined all the objects that contain the mouse position in step 410, the applet performs in step 414 all actions which are specified by behaviors which match the particular mouse event that occurred. In step 418, the applet notes any state changes caused by the actions performed. In step 422, the applet notifies any structures of state changes of objects in that structure. In step 430, the events which are triggered by the state changes noted in step 418 or by state changes that occurred in the containers which were notified in step 422 or elements in those containers are determined and control is transferred to state 408 where the triggered events are processed. In one embodiment, when such a triggered event occurs, control is transferred directly from state 408 to step 414 since step 410, looking up the objects which contain the mouse position, is not necessary for triggered events.<sup>3</sup>

This figure shows that when a user event occurs, such as a mouse click, state changes in the objects are noted (step 418) and the containers are notified of state changes (step 422), which can cause other events to be triggered (step 430 and "triggered event" exit from process events, step 408). When a screen change is necessary ("redraw event" exit from step 408), the method will "draw all visible objects" (step 408). Even more specifically, **Cook** states that "*all visible*

---

<sup>3</sup> Cook, column 9, line 54 through column 10, line 65, emphasis added

*objects, that is all objects which are not tagged as hidden, are redrawn*"<sup>4</sup>. Thus, each time **Cook** encounters a user event that causes a change to the screen, the entire page is re-rendered. This contrasts sharply with the invention recited in claim 1, in which differential rendering information, i.e., the difference between the existing display and the desired display, already exists and changes can be effected by applying the differential image information in order to create a composite image.

The rejection of claim 1 is therefore overcome. Independent claims 9, 17, 25, 27, and 30 are rejected for reasons similar to claim 1 and all contain a similar feature, so the rejection of these claims is also overcome. The independent claims that are listed above each contain features that are patentably distinct over **Cook** and their rejection is overcome. Therefore, the rejection of dependent claims 2-4, 18-20, 28-29, and 31-33 is also overcome.

#### **Claims 5, 13, and 21**

Independent claims 5, 13, and 21 have now been amended to more specifically recite the rendering information contained in the graphical image file. Exemplary claim 5 now recites:

5. (Amended) A method comprising:
  - receiving a graphical image file containing a plurality of graphical images and a respective event parameter corresponding to each of said plurality of graphical images, wherein each of said plurality of graphical images comprises respective rendering information;
  - initially displaying said respective rendering information for a first graphical image of said plurality of graphical images;
  - receiving an event corresponding to one of said plurality of event parameters; and
  - in response to said step of receiving an event, displaying respective rendering information for a second graphical image of said plurality of graphical images, said second graphical image having a respective event parameter corresponding to said event, wherein said respective rendering information for said second graphical image comprises differential image information that is combined with respective rendering information for said first graphical image to generate a composite image.

In addition to the features recited in claim 1, exemplary claim 5 recites that the graphical image file contains a "*plurality of graphical images*" and that rendering information for the first graphical image is combined with respective rendering information for the second graphical image to create a composite image. In contrast, **Cook** re-renders the page each time a change is

---

<sup>4</sup> **Cook**, column 10, lines 30-34

made. Exemplary claim 5 also emphasizes an additional difference between the presently claimed invention and **Cook**. **Cook** contains multiple objects that are manipulated, one for each item that can appear on the page. Each of these objects must be dealt with separately and can require the use of different image resources (and non-image resources, such as when sounds are used) when these objects are played. In contrast, claim 5 recites that the multiple graphical images are contained in a single graphical image file. There is only a single object that is dealt with, so only a single image resource is needed. Additionally, the claimed graphical image file requires less bandwidth to use. Thus, claim 5 provides patentable distinctions over **Cook** and the rejection is overcome for claim 5. Additionally, claims 13 and 21 are rejected for reasons similar to claim 5 and contain similar features, so the rejection of these claims are also overcome. Additionally, the rejection of claims 6-8, 14-16, and 22-24, each of which depends from one of claims 5, 13, and 21, are also overcome.

#### **Claim 26**

Independent claim 26 has been amended to recite, in addition to distinctions discussed above, that *“when a disposal method flag associated with said first graphical image is set to a value corresponding to “do not dispose”, said first and second graphical images are aggregated to generate a composite image”*. **Cook** does not anywhere disclose a disposal method flag that can be used to aggregate images to create a composite image. Therefore, the rejection of claim 26 is overcome.

Therefore, the rejection of claims 1-6, 9-14, 17-22, and 25-32 under 35 U.S.C. § 102(e) has been overcome.

Furthermore, **Cook** does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. **Cook** utilizes a different method to provide an interactive web page by re-drawing pages whenever the state of objects on the page changes (e.g., the state changes from “shown” to “not shown” or vice versa). Absent the examiner pointing out some teaching or incentive to modify **Cook** to implement any of (a) the sending of differential rendering information, rather than completely re-rendering, (b) the incorporation of all rendering information for the interactive page into a single object, and (c) the use of a flag to indicate “do not dispose” for current information, one of ordinary skill in the art would not be led to modify **Cook** to reach the present invention when the reference is examined as a whole. Absent some

teaching, suggestion, or incentive to modify **Cook** in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

V. **35 U.S.C. § 103, Obviousness: Claims 7-8, 15-16, and 23-24**

Claims 7-8, 15-16, and 23-24 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over **Cook**. This rejection is respectfully traversed.

Claim 7-8, 15-16, and 23-24 are all dependent claims whose independent claims have been discussed above. Since the independent claims have now been shown to contain patentable distinctions over **Cook**, these dependent claims also contain patentable distinctions over this reference.

Additionally, these claims show further features not shown or suggested by **Cook**. For example, claim 7 recites "*wherein said step of displaying respective rendering information for said second graphical image comprises the steps of: sequentially bypassing a graphical image in said plurality of graphical images while a count value does not equal said event parameter value; and if said count value equals said event parameter value, displaying respective rendering information of a current graphical image in sequence in said plurality of graphical images*".

The rejection states:

**Cook** describes a user input event of a mouse click on a web page at a certain position on the screen. The position of the cursor when the mouse click occurs is used as a value to decide which objects on the screen may require updating. As the software program searches through all the objects on the screen it will sequentially bypass sets of graphical images associated with objects that don't have an image to be displayed associated with the position of the cursor when the mouse was clicked. When the program find objects that have images to be displayed that match the position of the cursor when the mouse was clicked then these images are rendered on the screen column 10 lines 35-65.<sup>5</sup>

In part, the cited excerpt of **Cook** states the following: "[in] one embodiment, the user input is a mouse and so the applet will lookup all objects that contain the mouse position which is associated with the user input event"<sup>6</sup>. Here, **Cook** is searching through multiple objects to find ones that correspond to the current event. However, one of ordinary skill in the art would

---

<sup>5</sup> Office action of 10/04/05, page 12, lines 6-14

<sup>6</sup> **Cook**, column 10, lines 37-40



not consider that conducting a search through multiple objects is the same as sequentially bypassing graphical images. Thus, **Cook** neither discloses nor suggests this feature.

Therefore, the rejection of claims 7-8, 15-16, and 23-24 under 35 U.S.C. § 103(a) has been overcome.

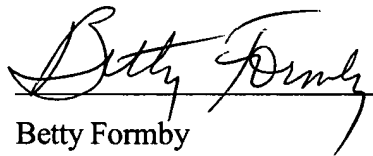
**VI. Conclusion**

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: December 22, 2005

Respectfully submitted,

A handwritten signature in cursive script, reading "Betty Formby", written over a horizontal line.

Betty Formby  
Reg. No. 36,536  
Yee & Associates, P.C.  
P.O. Box 802333  
Dallas, TX 75380  
(972) 385-8777  
Agent for Applicants